IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
YANNICK TEGLIA) Group Art Unit: 2137
Serial No.: 09/738,893)
Filed: December 15, 2000) Examiner: N. Khoshnoodi
)

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Claims 1, 2, 14, 22, and 25-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Iida in view of Pfab. Claims 3-7, 9-13, 16, 17, 19-21, and 24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Iida in view of Pfab and Menezes et al.

The present invention provides a method for secured transfer through a data bus that is connected between first and second memories. An N-byte data element is provided in the first memory, and the value of at least one parameter of a transfer rule is randomly chosen before a transfer of the N-byte data element. The transfer rule defines the order in which the bytes of the N-byte data element are successively transferred through the data bus. The N bytes of the data element are successively transferred byte-by-byte through the data bus to the second memory in the order specified by the transfer rule, with each of the N bytes transiting once and only once through the data bus. Thus, in this transfer method, the bytes of the data element stored in a first memory are successively transferred byte-by-byte to the second memory, and the transfer rule with at least one randomly chosen parameter is used to select the byte to be transferred at each successive transfer of a byte. The transfer rule, randomly chosen parameter(s), and successive byte-by-byte transfer of the data element operate together so that the N bytes of the data element are not transferred in the same successive byte order for each transfer of that data element.

Neither Pfab nor Iida, or a combination of the two, discloses a method for secured transfer in which the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred, is randomly chosen before a transfer of the data element, and the N bytes of the data element are successively transferred byte-by-byte through a data bus in the order specified by the transfer rule.

Applicant traverses the Examiner position that, while Iida does not disclose randomly choosing the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred through the data bus, before a transfer of the N-byte data element, the Pfab reference makes up for this deficiency.

In the present invention, the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte, is randomly chosen before a successive byte-by-byte transfer of the data element in the byte order that is specified by the transfer rule. More specifically, a transfer rule defines the order

in which the N bytes of an N-byte data element are successively transferred through a data bus, and the value of parameter(s) of the transfer rule are randomly chosen before a transfer. The N bytes of the data element are successively transferred byte-by-byte through the data bus in the byte order specified by the transfer rule, with each of the N bytes transiting once and only once through the data bus. Thus, the bytes of the N-byte data element are successively transferred byte-by-byte, and the transfer rule with at least one randomly chosen parameter is used to select the byte to be transferred at each successive transfer of one byte of the N bytes. Thus, the N bytes of the data element are not successively transferred in the same order for each transfer.

In contrast, Pfab discloses a data processing circuit in which data is stored in memory and transferred through the data bus in an encoded format. In two embodiments, each memory 102-105 stores encoded data, and this encoded data is transferred through the data bus 106. The microprocessor 101 includes an encoding module 107 that decodes the encoded data received from the data bus 106, and encodes data to be sent each memory 102-105 through the data bus 106. In another embodiment, each memory 2-5 stores encoded data, and this data is transferred through the data buses 6-15 at least partially encoded. The encoded data is partially decoded by an associated encoding module 20-22 and then completely decoded by an encoding module 35 of the microprocessor. Thus, Pfab teaches data processing circuits in which data is protected by modifying (i.e., encoding) each data byte stored in memory and transferred through the data bus.

While Pfab teaches modifying each data unit (e.g., byte or "word") that is transferred through the data bus at one time, Pfab does not teach or suggest modifying the order in which these data units (e.g., bytes or "words") are successively transferred through the data bus. Pfab teaches altering each data unit that transits through the data bus, but does not teach or suggest altering the order in which multiple data units are successively transferred through the data bus.

The Examiner states that "Pfab's disclosure of altering the significance of the different bits is equivalent to that of defining the order in which the bytes are transferred because of the fact that 1 byte consists of eight bits." Pfab discloses that an encoding module encodes data traffic on the data bus, and that the encoding can be performed by altering the significance of individual data bits. See Pfab at 6:44-57. Pfab teaches using hardware encoding to alter the significance of individuals bits by changing "low" bits of a data unit to be transferred to "high" bits when the data unit is transferred on the data bus. See Pfab at 3:40-49. Thus, Pfab discloses changing the bit values of each data unit that transits through the data bus. In other words, the values of bits in one data unit (i.e., the amount of data that is transported simultaneously on the

data bus) are modified.

In contrast, in the present invention, the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte through a data bus, is randomly chosen before a successive byte-by-byte transfer of the data element through a data bus in the byte order that is specified by the transfer rule. The claims recite: 1) successive byte-by-byte transfer of an N-byte data element (i.e., one byte is transferred, then another byte, and so on); 2) the order in which the bytes of the element data are successively transferred is defined by a transfer rule having randomly chosen parameter(s); and 3) each byte of the N-byte data element is transferred once and only once during the transfer of the N-byte data element. This successive transfer of the N bytes of the data element byte-by-byte through the data bus in a byte order that is specified by a transfer rule having randomly chosen parameter(s) is used to alter the order in which data units (i.e., bytes in this case) are transferred data unit-by-data unit through the data bus. This is fundamentally different than Pfab's teaching of altering the values of the bits in each data unit that is transferred.

Further, "the fact that 1 byte consists of eight bits" is irrelevant. Pfab teaches altering each data unit before that data unit is transferred on the data bus, while the claims recite altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. Regardless of the number of bits or bytes in a data unit, Pfab only teaches altering each data unit that is successively transferred on the data bus, not changing the order in which data units are successively transferred data unit-by-data unit through a data bus. In other words, if an N-byte data element is transferred byte-by-byte (as recited in the claims) in Pfab, each byte that is transferred will be changed, but the N bytes are always transferred in the same byte order.

The Examiner also states that "Pfab's disclosure of determining which bit lines should be used also determines the byte order in which the N-byte data element will progress on the bus." Pfab never teaches "determining which bit lines should be used". In Pfab, all of the bit lines of the data bus are always used. Pfab actually teaches that an encoding module can encode data traffic on a data bus by interchanging individual bit lines of the data bus. See Pfab at 6:44-57. Pfab's disclosure of "interchanging individual data bits" entails changing the order of the bits of each data unit when that data unit transits through the data bus. That is, the bit order within one data unit is modified. Pfab does not teach or suggest changing the order in which multiple data units are successively transferred data unit-by-data unit through the data bus.

In contrast, in the present invention, the value of at least one parameter of a transfer rule,

which defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte through a data bus, is randomly chosen before a successive byte-by-byte transfer of the data element through a data bus in the byte order that is specified by the transfer rule. This successive transfer of the N bytes of the data element byte-by-byte through the data bus in a byte order that is specified by a transfer rule having randomly chosen parameter(s) is used to alter the order in which data units (i.e., bytes in this case) are transferred data unit-by-data unit through the data bus. This is fundamentally different than Pfab's teaching of altering the bit order in each data unit that is transferred through the data bus.

Pfab teaches altering each data unit that is transferred on the data bus, while the claims recite altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. Pfab does not teach or suggest changing the order in which data units are successively transferred data unit-by-data unit through a data bus. If an N-byte data element is transferred byte-by-byte (as in the claims) using the transfer method of Pfab, each byte that is transferred will be changed, but the N bytes will always be transferred in the same byte order.

The Examiner also states that randomly selecting a key in Pfab is equivalent to randomly choosing a parameter of the transfer rule. Pfab teaches using an encoding module to encode data traffic on a data bus by changing each data unit that transits through the data bus, and randomly selecting the encoding key. In contrast, in the present invention, there is randomly selected the value of at least one parameter of a transfer rule that defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte, before a successive byte-by-byte transfer of the data element in the byte order that is specified by the transfer rule. The randomly chosen parameter(s) of the transfer rule is used to alter the order in which data units are transferred. This is fundamentally different than randomly choosing a key for altering each data unit that is transferred through the data bus.

Pfab teaches randomly choosing a key for altering each data unit that is transferred, while the claims recite randomly choosing a parameter(s) of the transfer rule. Pfab does not teach or suggest randomly choosing a parameter(s) of a transfer rule for changing the order in which data units are successively transferred. In other words, if an N-byte data element is transferred byte-by-byte (as recited in the claims) in Pfab, each byte that is transferred will be changed, but the N bytes will always be transferred in the same byte order regardless of the randomly selected key.

The Examiner also states that "Pfab teaches that the data element is transferred byte-by-byte because a sequence of 8 bits is equivalent to one byte and there is an operating

module that can influence the encoding using different conversion methods." As explained above, Pfab teaches using an encoding module to encode data traffic on a data bus by changing each data unit that transits through the data bus, using different methods for altering each data unit (such as encoding, changing bit values, and changing bit order). However, all of the disclosed encoding or conversion methods entail changing each individual data unit.

In contrast, in the present invention, a transfer rule defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte, and the data element is successively transferred byte-by-byte in the byte order that is specified by the transfer rule. The transfer rule is used to alter the order in which data units are transferred data. This is fundamentally different than Pfab's teaching of altering each data unit that is transferred through the data bus. Further, the fact that eight bits is equal to one byte is irrelevant to the fundamental difference between Pfab and the transfer method recited in the claims, as explained above.

In Pfab, a data element having four data units will always be successively transferred data unit-by-data unit in the same order: D1, D2, D3, then D4. Pfab secures each data element by altering it, not by changing the transfer order of the data units. In the present invention, four such data units D1-D4 of a data element are not always successively transferred in the same order. Instead, the data units are successively transferred in an order defined by a transfer rule with random parameter(s). The same data element may be transferred first in the order D3, D2, D1, D4, and then in the order D2, D3, D4, D1, and then in the order D4, D3, D2, D1.

Pfab teaches altering each transferred data unit, not randomly choosing a parameter(s) of a transfer rule for changing the order in which data units are successively transferred data unit-by-data unit through a data bus. Therefore, the rejections should be withdrawn.

Additionally, claims 1, 14, and 22 were rejected for double patenting. In the cited patent, each byte is transferred a random number of times, and a random number of total bytes are transferred. Here, each byte of the data is transferred exactly one time on the data bus. Also, in the cited patent, the place value of the byte to be transferred is randomly chosen. Here, a transfer rule defines the order in which the bytes are successively transferred, at least one parameter of the transfer rule is randomly chosen, and this same value is used for all bytes of the data element. Therefore, this rejection should be withdrawn.

Respectfully submitted,

Date: December 10, 2007

By: /Stephen Bongini/

Stephen Bongini
Registration No. 40,917

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450,

on 12/10/07.	Stephen Bongini	/Stephen Bongini/
Date of Deposit	Applicant, Assignee, or Representative	Signature